Filed: June 23, 2003
Phone: (215) 568-3100

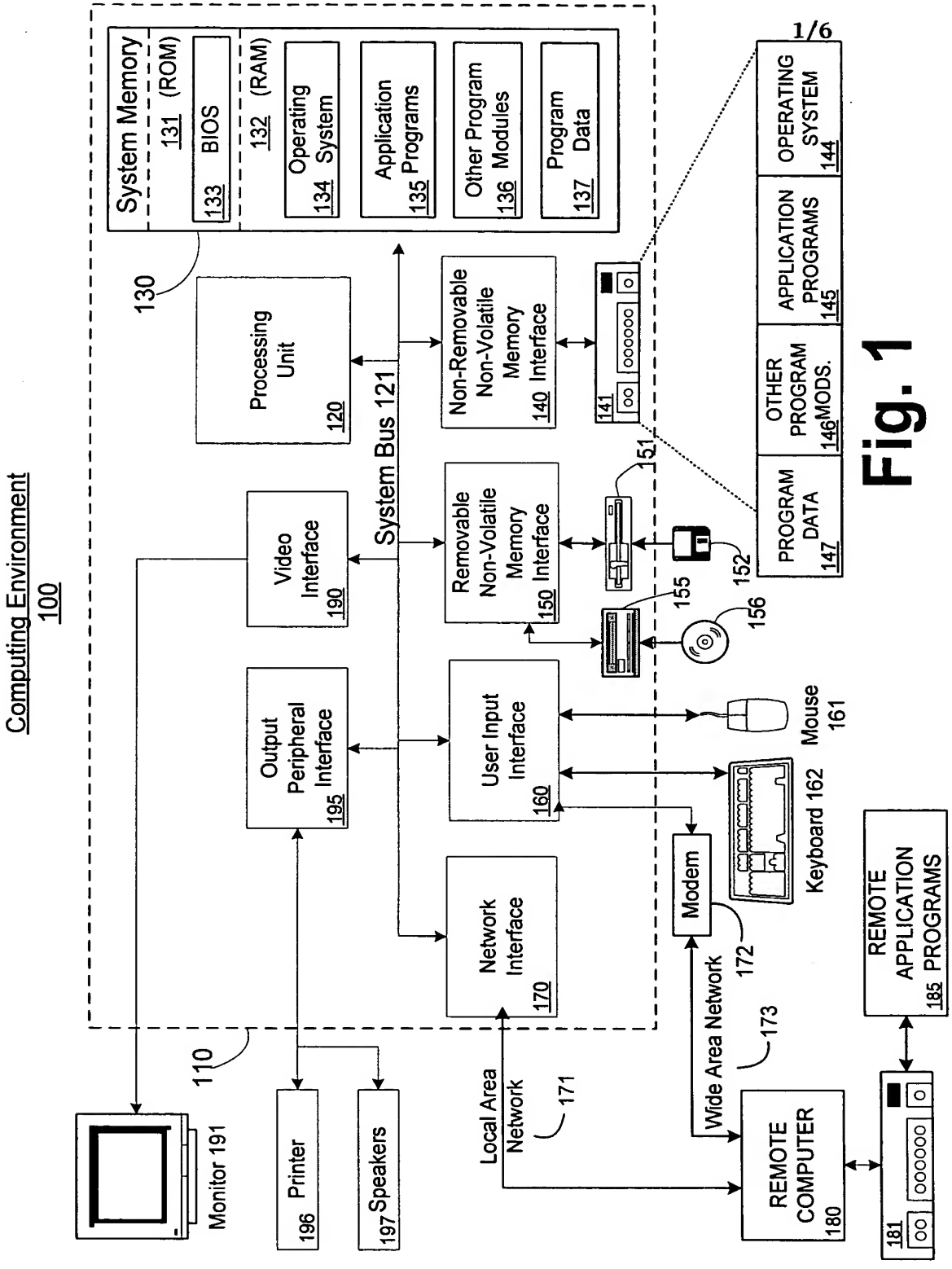

Fig. 1

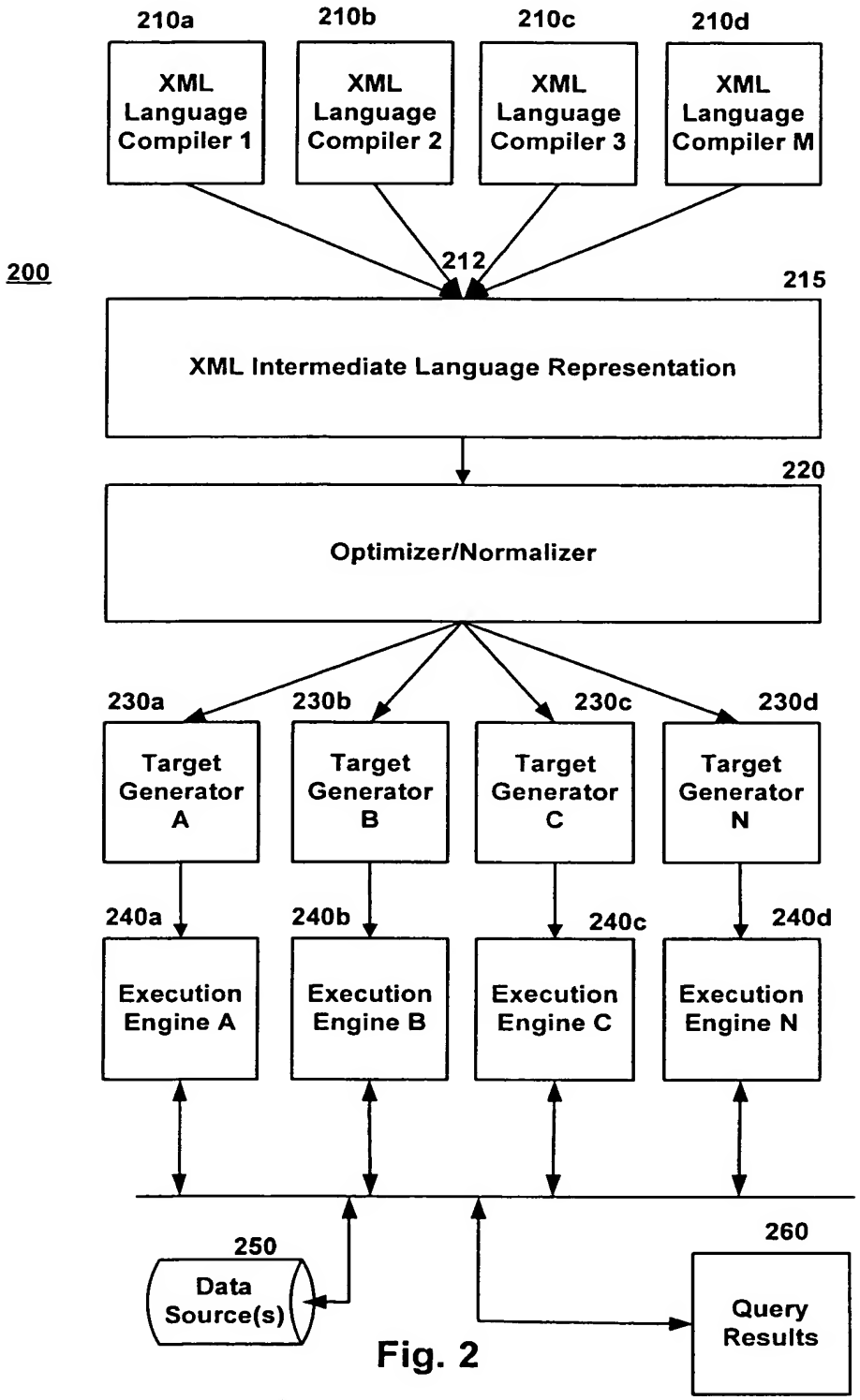
200
210a
XML Language Compiler 1
210b
XML Language Compiler 2
210c
XML Language Compiler 3
210d
XML Language Compiler M
212
215
XML Intermediate Language Representation
220
Optimizer/Normalizer
230a
Target Generator A
230b
Target Generator B
230c
Target Generator C
230d
Target Generator N
240a
Execution Engine A
240b
Execution Engine B
240c
Execution Engine C
240d
Execution Engine N
250
Data Source(s)
260
Query Results

Fig. 2

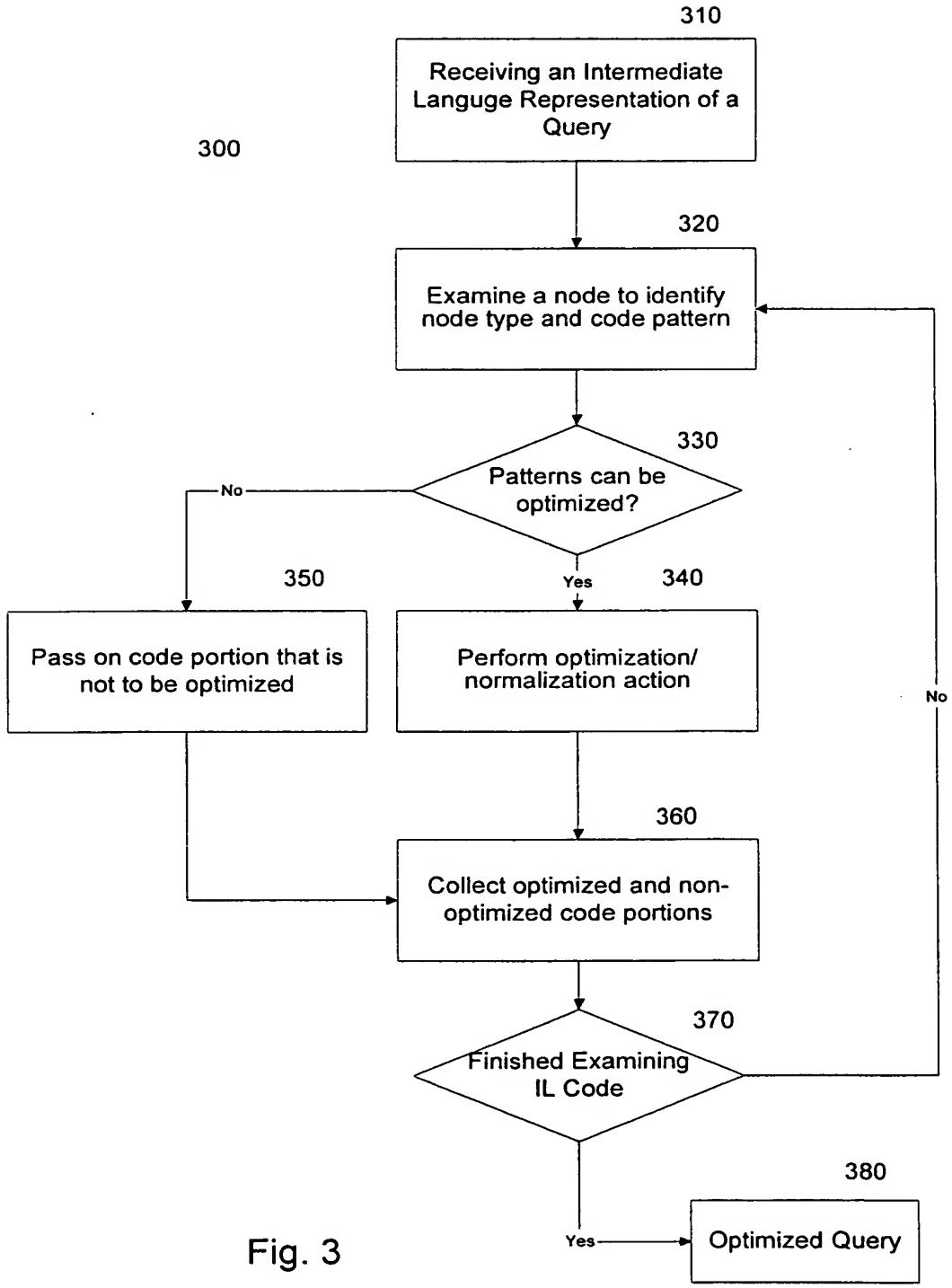
300
310
Receiving an Intermediate Languge Representation of a Query
320
Examine a node to identify node type and code pattern
330
Patterns can be optimized?
No
Yes
340
Perform optimization/ normalization action
350
Pass on code portion that is not to be optimized
360
Collect optimized and non-optimized code portions
370
Finished Examining IL Code
No
Yes
380
Optimized Query

Fig. 3

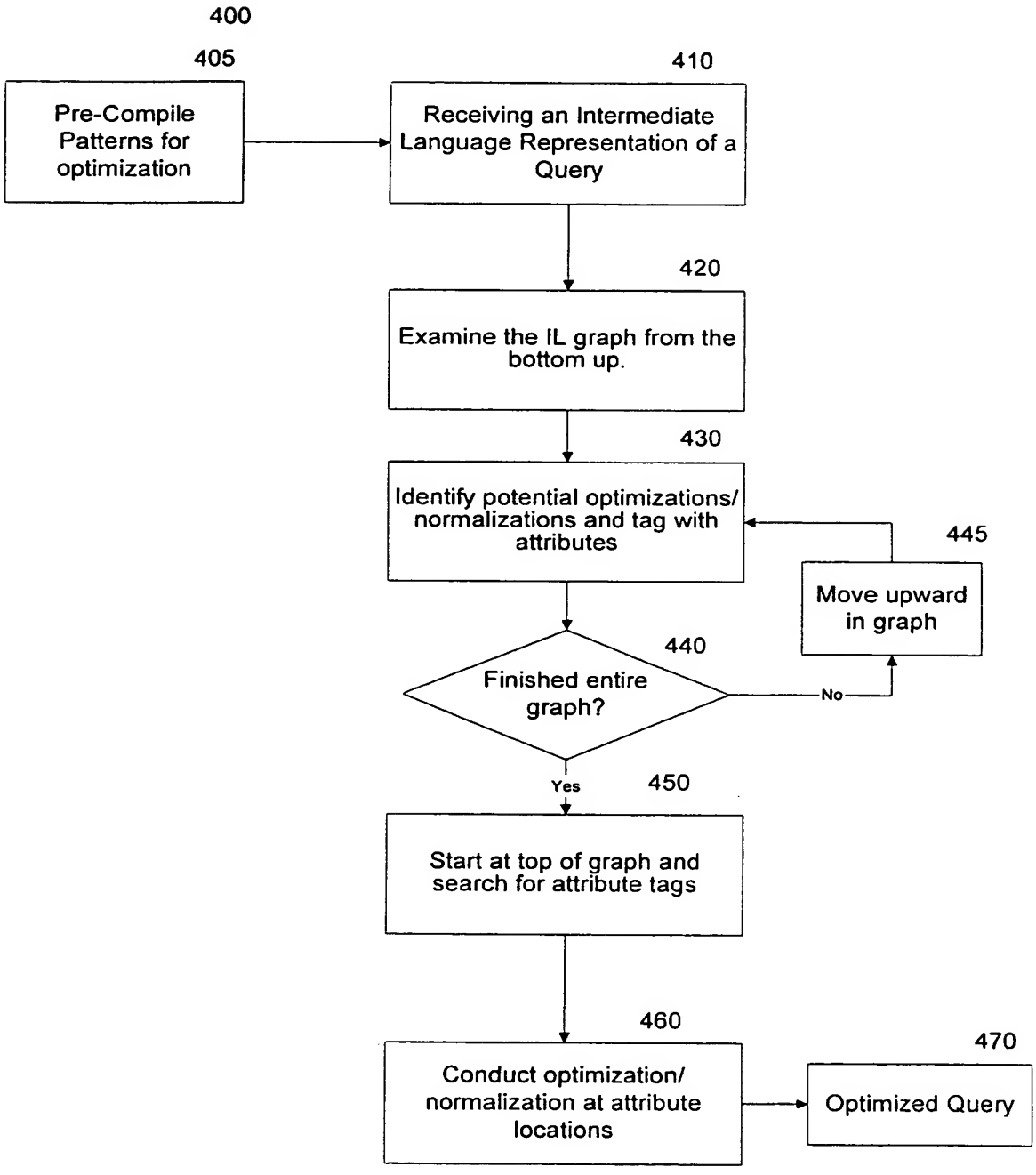
400
405
Pre-Compile Patterns for optimization
410
Receiving an Intermediate Language Representation of a Query
420
Examine the IL graph from the bottom up.
430
Identify potential optimizations/ normalizations and tag with attributes
445
Move upward in graph
440
Finished entire graph?
No
Yes
450
Start at top of graph and search for attribute tags
460
Conduct optimization/ normalization at attribute locations
470
Optimized Query

Fig. 4

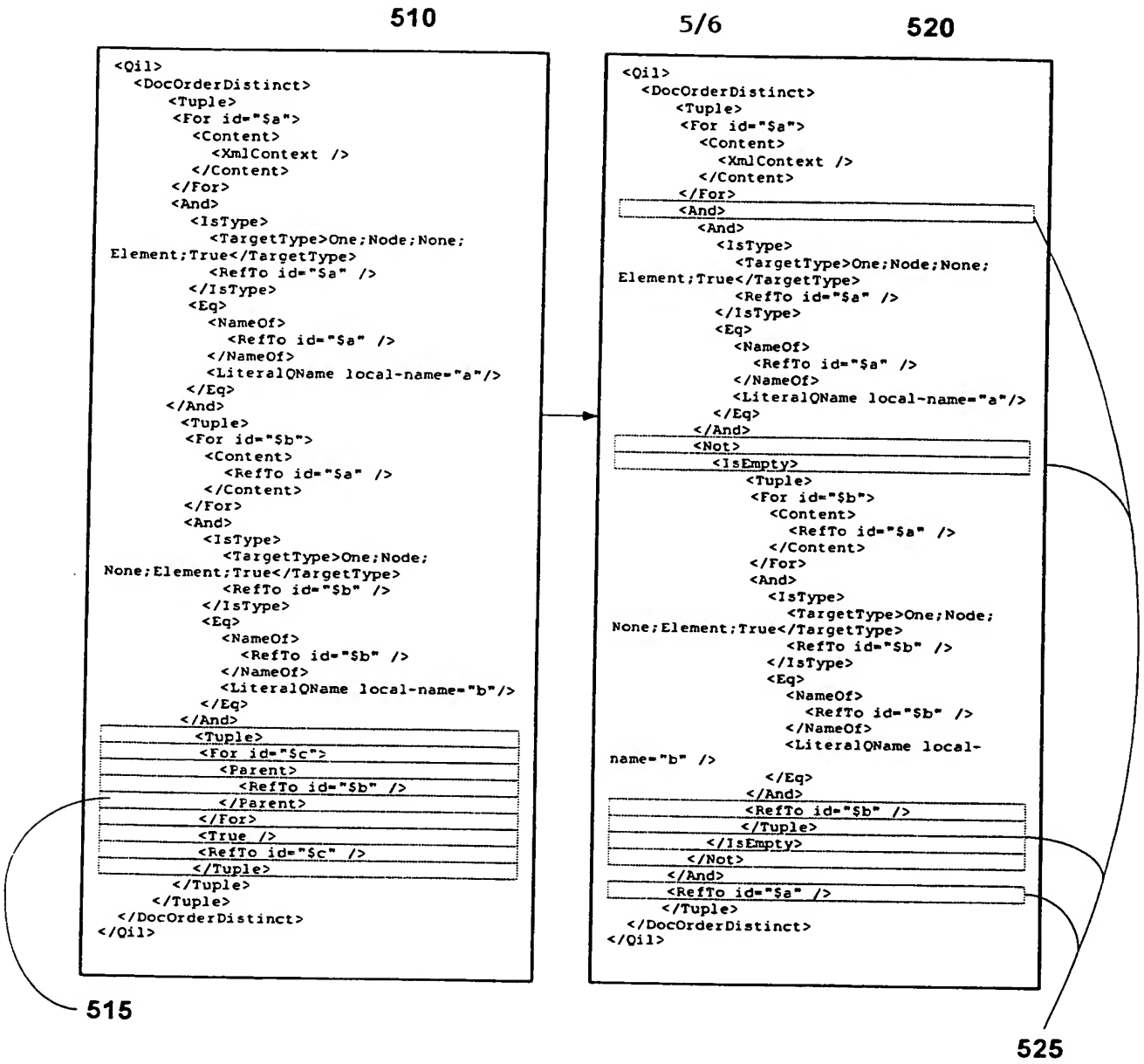
510
5/6
520
<Qil>
<DocOrderDistinct>
<Tuple>
<For id="$a">
<Content>
<XmlContext />
</Content>
</For>
<And>
<IsType>
<TargetType>One;Node;None;Element;True</TargetType>
<RefTo id="$a" />
</IsType>
<Eq>
<NameOf>
<RefTo id="$a" />
</NameOf>
<LiteralQName local-name="a"/>
</Eq>
</And>
<Tuple>
<For id="$b">
<Content>
<RefTo id="$a" />
</Content>
</For>
<And>
<IsType>
<TargetType>One;Node;None;Element;True</TargetType>
<RefTo id="$b" />
</IsType>
<Eq>
<NameOf>
<RefTo id="$b" />
</NameOf>
<LiteralQName local-name="b"/>
</Eq>
</And>
<Tuple>
<For id="$c">
<Parent>
<RefTo id="$b" />
</Parent>
</For>
<True />
<RefTo id="$c" />
</Tuple>
</Tuple>
</Tuple>
</DocOrderDistinct>
</Qil>
<Qil>
<DocOrderDistinct>
<Tuple>
<For id="$a">
<Content>
<XmlContext />
</Content>
</For>
<And>
<And>
<IsType>
<TargetType>One;Node;None;Element;True</TargetType>
<RefTo id="$a" />
</IsType>
<Eq>
<NameOf>
<RefTo id="$a" />
</NameOf>
<LiteralQName local-name="a"/>
</Eq>
</And>
<Not>
<IsEmpty>
<Tuple>
<For id="$b">
<Content>
<RefTo id="$a" />
</Content>
</For>
<And>
<IsType>
<TargetType>One;Node;None;Element;True</TargetType>
<RefTo id="$b" />
</IsType>
<Eq>
<NameOf>
<RefTo id="$b" />
</NameOf>
<LiteralQName local-name="b" />
</Eq>
</And>
<RefTo id="$b" />
</Tuple>
</IsEmpty>
</Not>
</And>
<RefTo id="$a" />
</Tuple>
</DocOrderDistinct>
</Qil>
515
525

Fig. 5

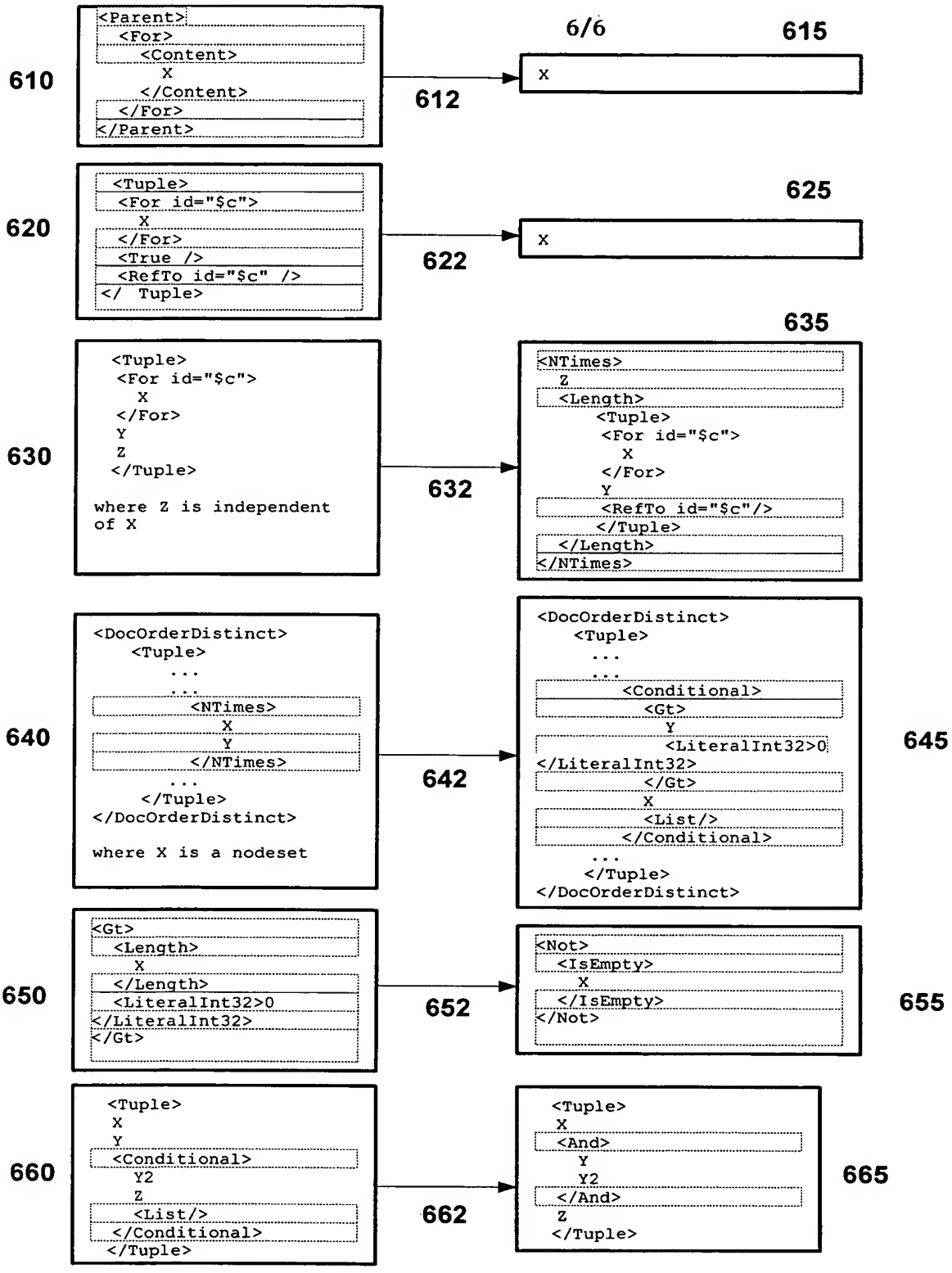
6/6
610
<Parent>
<For>
<Content>
X
</Content>
</For>
</Parent>
612
615
X
620
<Tuple>
<For id="$c">
X
</For>
<True />
<RefTo id="$c" />
</ Tuple>
622
625
X
630
<Tuple>
<For id="$c">
X
</For>
Y
Z
</Tuple>
where Z is independent of X
632
635
<NTimes>
Z
<Length>
<Tuple>
<For id="$c">
X
</For>
Y
<RefTo id="$c"/>
</Tuple>
</Length>
</NTimes>
640
<DocOrderDistinct>
<Tuple>
...
...
<NTimes>
X
Y
</NTimes>
...
</Tuple>
</DocOrderDistinct>
where X is a nodeset
642
<DocOrderDistinct>
<Tuple>
...
...
<Conditional>
<Gt>
Y
<LiteralInt32>0
</LiteralInt32>
</Gt>
X
<List/>
</Conditional>
...
</Tuple>
</DocOrderDistinct>
645
650
<Gt>
<Length>
X
</Length>
<LiteralInt32>0
</LiteralInt32>
</Gt>
652
<Not>
<IsEmpty>
X
</IsEmpty>
</Not>
655
660
<Tuple>
X
Y
<Conditional>
Y2
Z
<List/>
</Conditional>
</Tuple>
662
<Tuple>
X
<And>
Y
Y2
</And>
Z
</Tuple>
665

Fig. 6